

## **Claims**

1. (Currently Amended) A method of generating identifier data for persistently identifying a user interface element of interest in a graphical user interface of a source computer program, the method comprising:

receiving data indicative of the user interface element of interest from a first software component; and

in response to receiving the data indicative of the user interface element of interest, generating an element path identifier of the user interface element of interest for persistently and uniquely identifying the user interface element of interest and returning at least the unique element path identifier to the first software component;

wherein persistently and uniquely identifying the user element of interest comprises persistently and uniquely identifying the user interface element of interest across reboots of the computer running the source computer program.

2. (Original) The method of claim 1, wherein generating the element path identifier is implemented by a second software component communicative with the source computer program.

3. (Original) The method of claim 1, wherein generating an element path identifier comprises:

using a hierarchical tree structure representation of the graphical user interface to locate a leaf node related to the user interface element of interest;

storing exposed identifier information of the user interface element of interest in an element path identifier data structure; and

proceeding up the hierarchy of the tree structure representation to store the exposed identifier information of selected parent nodes of the user interface element of interest in the element path identifier data structure.

4. (Original) The method of claim 3, further comprising converting the element path identifier to a string type data structure.

5. (Original) The method of claim 3, further comprising receiving data indicative of a designated element path root node of the tree structure representation, wherein the step of proceeding up the hierarchy of the tree structure representation to store the exposed identifier information of selected parent nodes is continued only until the element path root node is reached.

6. (Original) The method of claim 3, wherein the exposed identifier information is a local alpha numeric identifier not guaranteed to be unique, a class name, a module name associated with an application program or a sibling order.

7. (Original) The method of claim 3, further comprising:  
determining that at least one of the selected parent nodes is a root node of a strongly named branch portion of the tree structure representation, wherein the strongly named branch portion has at least one user interface element within a scope inside of which the at least one user interface element is guaranteed to be uniquely identifiable by a named branch element identifier;  
and

storing in the element path identifier data structure, the named branch element identifier for the at least one user interface element within the scope of the named branch portion.

8. (Original) The method of claim 7, further comprising:  
storing in the element path identifier a strong name associated with the strongly named branch along with the named branch element identifier.

9. (Original) The method of claim 7, wherein the named branch element identifier stored in the element path identifier data structure is for the user interface element of interest and no identifying information related to the parent elements within the scope are stored in the element path identifier data structure.

10-20. (Canceled)

21. (Currently Amended) At least one computer-readable medium having stored thereon computer-executable instructions related to a function responsive to a function call from a first software component, the function comprising:

an input parameter representing a user interface element of interest in a graphical user interface of a source computer program;

an output parameter representing an element path identifier for persistent unique identification of the user interface element of interest; wherein the element path identifier comprises a hierarchical path of inheritance from the user interface element of interest to a parent root element; and

executable software for receiving the input parameter representing an user interface element of interest and in response, generating the output parameter representing an element path identifier of the user interface element of interest such that the output parameter represents an identifier capable of persistently identifying the user interface element of interest across different builds of the computer program.

22. (Original) The computer-readable medium of claim 21, wherein the element path identifier is a string type data structure.

23. (Original) The computer-readable medium of claim 21, wherein the element path identifier comprises exposed identifier information of component elements of an element path related to the element of interest.

24. (Original) The computer-readable medium of claim 21, wherein the element path identifier comprises class names of component elements of an element path of the element of interest.

25. (Original) The computer-readable medium of claim 21, wherein the element path identifier comprises a module name of an application program related to component elements of an element path of the element of interest.

26. (Original) The computer-readable medium of claim 21, wherein the element path identifier comprises sibling order data.

27. (Original) The computer-readable medium of claim 21, wherein generating the output parameter representing an element path identifier comprises determining that at least one node in a hierarchical tree structure representation of the graphical user interface of the source computer program is a root node of a strongly named branch portion of the tree structure representation, wherein the strongly named branch portion has at least one user interface element within a scope inside of which the at least one user interface element is guaranteed to be uniquely identifiable by a named branch element identifier, and the element path identifier comprises the named branch element identifier for the at least one user interface element within the scope of the named branch portion.

28. (Currently Amended) At least one computer-readable medium having stored thereon computer-executable instructions related to a function responsive to a function call from a first software component, the function comprising:

an input parameter representing an element path identifier of a target user interface element in a graphical user interface of a target computer program, wherein the element path identifier comprises a hierarchical path of inheritance from the target user interface to a parent root element;

an output parameter representing a location of the target user interface element within a hierarchical tree structure representation of the graphical user interface; and

executable software for receiving the element path identifier of the target user interface element and determining the output parameter representing the location of the target user interface element such that the output parameter represents an identifier capable of persistently identifying the target user interface element across different builds of the computer program.

29. (Original) The computer-readable medium of claim 28, wherein the element path identifier is a string type data structure.

30. (Original) The computer-readable medium of claim 28, wherein the element path identifier comprises exposed identifier information of component elements of an element path related to the element of interest.

31. (Original) The computer-readable medium of claim 28, wherein the element path identifier comprises class names of component elements of an element path of the element of interest.

32. (Original) The computer-readable medium of claim 28, wherein the element path identifier comprises a module name of an application program related to component elements of an element path of the element of interest.

33. (Original) The computer-readable medium of claim 28, wherein the element path identifier comprises sibling order data.

34. (Original) The computer-readable medium of claim 28, wherein hierarchical tree structure representation of the graphical user interface of the target computer program comprises at least one node that is root node of a strongly named branch portion of the tree structure representation, wherein the strongly named branch portion has at least one user interface element node within a scope inside of which the at least one user interface element node is guaranteed to be uniquely identifiable by a named branch element identifier and the input parameter representing the element path identifier comprises the named branch element identifier of the at least one user interface element guaranteed to be uniquely identifiable.

35. (Currently Amended) In a computer system running a computer program with a graphical user interface, a system for generating persistent, unique element path identifiers of elements of the graphical user interface and later searching for the elements of the graphical user interface using the element path identifiers, the system comprising:

an API module comprising a first set of APIs related to passing function calls for generating the element path identifiers and a second set of APIs related to passing function calls

for searching for the elements of the graphical user interface using the element path identifiers;  
and

an element path engine responsive to the function calls for generating the element path identifiers and to the function calls for searching for the elements of the graphical user interface using the element path identifiers, wherein the element path identifiers comprise information relating to parent elements of the elements of the graphical user interface such that the element path identifiers persistently identify the user interface elements across a build of the computer program.

36. (New) The method of claim 1, wherein persistently uniquely identifying the user element of interest further comprises persistently uniquely identifying the user interface element of interest across different builds of the computer program.

37. (New) The method of claim 21, wherein the output parameter further represents an identifier capable of persistently identifying the target user interface element across different builds of the computer program.

38. (New) The computer system of claim 35 wherein the element path identifier identifies the application in which the user interface element resides.

39. (New) The computer system of claim 35 wherein the element path identifier identifies the user interface platform used to create the user interface element.

40. (New) The computer system of claim 35 wherein the element path identifier identifies the application frame in which the user interface element resides.